BERKELEY LAB

HEP-CCE

Quantum Associative Memory in HEP Track Pattern Recognition

Illya Shapoval, Paolo Calafiura

Lawrence Berkeley National Laboratory

CHEP 2018, Sofia, Bulgaria
9-13 July, 2018

1.

Motivation

Memory in real-time pattern matching

	RAM	Associative Memory (CAM)	
		ASIC chips	Hopfield networks
Capacity[1] (patterns)	$\mathcal{O}(N/d)$	$\mathcal{O}(N/d)$	$\mathcal{O}(N)_{N\geq d}$: ≤kN, 0.138≤k≤2 [2]
Recall speed	$\mathcal{O}(N/d)$	$\mathcal{O}(1)$	$\Omega(N)_{N\geq d}$
Spurious memories	None	None	Possible (e.g., $\mathcal{O}(e^N)$ for a Hebb-type rule[3])
Recall of incomplete/noisy inputs	Yes/No	Yes/No	Yes/Yes
Power dissipation	Low	High	Low
Cost	Low	High	R&D

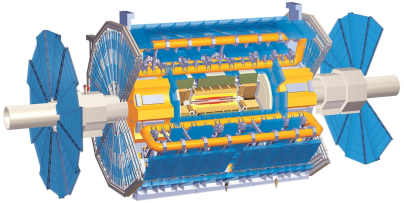
[1] **N** is the total number of elementary storage units in a device (memory cells/neurons), **d** - length of a pattern.

[2] B. Muller, J. Reinhardt, M.T. Strickland, Neural networks: An Introduction. Springer, 1995

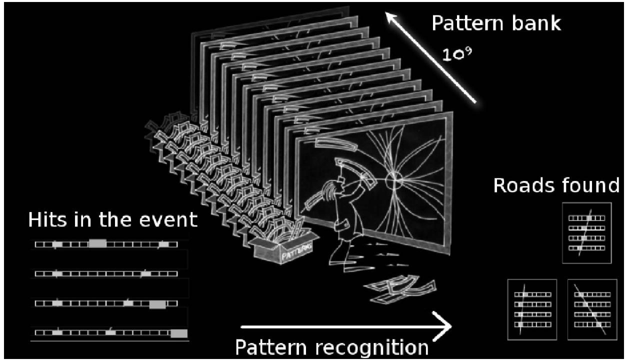
[3] J. Bruck, V. P. Roychowdhury. On The Number of Spurious Memories in the Hopfield Model. IEEE Trans. on Information Theory, V. 36, 2, 1990

An example: ATLAS Fast Tracker (FTK)

LHC Run 2 (2015) - Run 3 (2023)



A HARDWARE FOR REAL-TIME GLOBAL TRACK FINDING



Requirements:

- Input: 10^8 channels
- Latency: ~100 us
- Frequency: @100 kHz

Pattern recognition engine: **Associative Memory**

- Storage: $8 \cdot 10^3$ AM custom ASIC chips
- Power: ~32 kW (+ cooling)
- Capacity: **10^9** track patterns
- Latency: average ~50 us, max ~180 us

40 MHz
[50 TB/s]

Level 1
hardware-based
(~2.5 us)

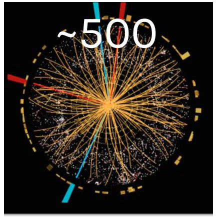
100 kHz
[150 GB/s]

FTK

High Level Trigger
software-based
(~seconds)

1 kHz
[~ GB/s]

Scalability of Associative Memory

Experiment	LHC Run 2-3	HL-LHC (2026)	HE-LHC (2030s)
LHC Luminosity ($cm^{-2}s^{-1}$)	$\sim 10^{34}$	$\sim 10^{35}$	$\sim 10^{36}$
Tracks/event	 ~500	 ~5,000	~50,000
AM Capacity* (patterns)	10^{9}	$[8 - 16] \cdot 10^{9}$	?
AM Storage* (AM chips)	$8 \cdot 10^{3}$	$[2 - 4] \cdot 8 \cdot 10^{3}$	?
Density* (patterns/chip)	128k (65 nm)	~512k (28 nm)	?

* Required by ATLAS physics and detector granularity

2.

Quantum Associative Memory

Quantum Memory

- Represent pattern $\xi^i \equiv (\xi_1, \xi_2, \ldots, \xi_d)$ by a **basis state** in the Hilbert space of *d* quantum information units:

$$|\xi^i\rangle \equiv |\xi_1, \xi_2, \ldots, \xi_d\rangle$$

- Represent Ξ - a set of *N* patterns - as **superposition** of the basis states:

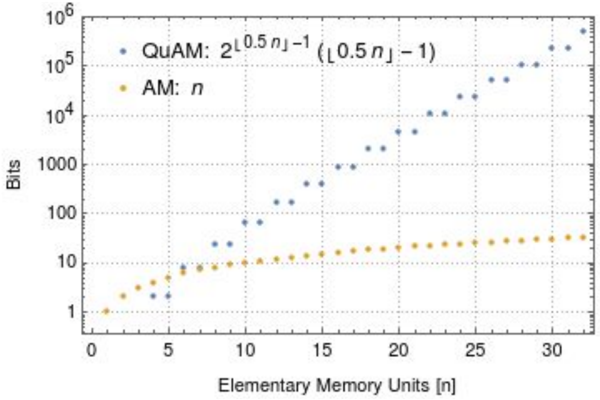
$$|\Xi\rangle = \sum_1^N \alpha_i |\xi^i\rangle, \qquad \alpha_i \in \mathbb{C} \ \wedge \ \sum_1^N |\alpha_i|^2 = 1$$

Estimate

Absolute QuAM capacity

QuAM features exponential storage capacity of *2^d* and requires *$2(d+1)$* qubits to operate[1,2].

Length of detector hit identifier (bits)	8	16	32
Length of binary track pattern (bits) [3]	64	128	256
QuAM register (qubits)	130	258	514
QuAM capacity (patterns)	**$\sim 10^{19}$**	**$\sim 10^{38}$**	**$\sim 10^{77}$**



[1] C.A Trugenberger, Probabilistic Quantum Memories. Phys Rev. Lett. Vol 87, 6 (2001)
[2] *d* is the pattern length
[3] 8 logical layers of the Inner Tracker

QuAM storage protocol

A quantum circuit implementing the iterative part of the storage protocol [1].

$$1.\ |\psi_0^1\rangle = |p_1^1, ..., p_d^1; 01; 0_1, ..., 0_d\rangle$$

$$2.\ |\psi_1^i\rangle = \prod_{j=1}^{d} {}^{2c}\hat{X}_{p_j^i u_2 m_j} |\psi_0^i\rangle$$

$$3.\ |\psi_2^i\rangle = \prod_{j=1}^{d} \hat{X}_{m_j} {}^{1c}\hat{X}_{p_j^i m_j} |\psi_1^i\rangle$$

$$4.\ |\psi_3^i\rangle = {}^{dc}\hat{X}_{m_1 \dots m_d u_1} |\psi_2^i\rangle$$

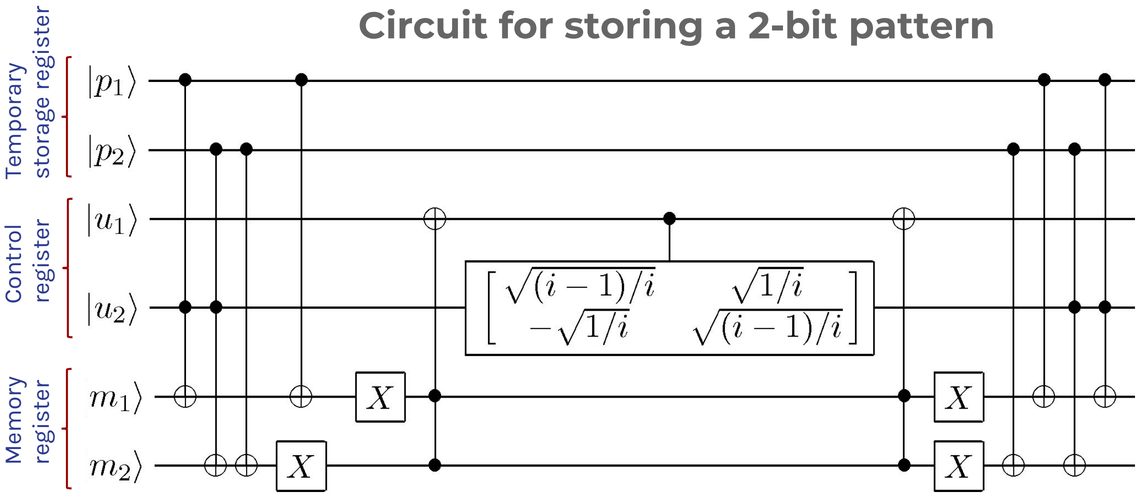
$$5.\ |\psi_4^i\rangle = {}^{1c}\hat{S}_{u_1 u_2}(p + 1 - i)|\psi_3^i\rangle$$

$$6.\ |\psi_5^i\rangle = {}^{dc}\hat{X}_{m_1 \dots m_d u_1} |\psi_4^i\rangle$$

$$7.\ |\psi_6^i\rangle = \prod_{j=d}^{1} {}^{1c}\hat{X}_{p_j^i m_j} \hat{X}_{m_j} |\psi_1^i\rangle$$

$$= \frac{1}{\sqrt{p}} \sum_{k=1}^{i} |p^i; 00; p^k\rangle + \sqrt{\frac{p-i}{p}} |p^i; 01; p^i\rangle$$

$$8.\ |\psi_7^i\rangle = \prod_{j=d}^{1} {}^{2c}\hat{X}_{p_j^i u_2 m_j} |\psi_6^i\rangle$$



[1] C.A Trugenberger, Probabilistic Quantum Memories. Phys Rev. Lett. Vol 87, 6 (2001)

QuAM storage protocol

A quantum circuit implementing the iterative part of the storage protocol [1].

1. $|\psi_0^1\rangle = |p_1^1, ..., p_d^1; 01; 0_1, ..., 0_d\rangle$

2. $|\psi_1^i\rangle = \prod_{j=1}^{d} {}^{2c}\hat{X}_{p_j^i u_2 m_j} |\psi_0^i\rangle$

3. $|\psi_2^i\rangle = \prod_{j=1}^{d} \hat{X}_{m_j} \, {}^{1c}\hat{X}_{p_j^i m_j} |\psi_1^i\rangle$

4. $|\psi_3^i\rangle = {}^{dc}\hat{X}_{m_1 \ldots m_d u_1} |\psi_2^i\rangle$

5. $|\psi_4^i\rangle = {}^{1c}\hat{S}_{u_1 u_2}(p+1-i) |\psi_3^i\rangle$

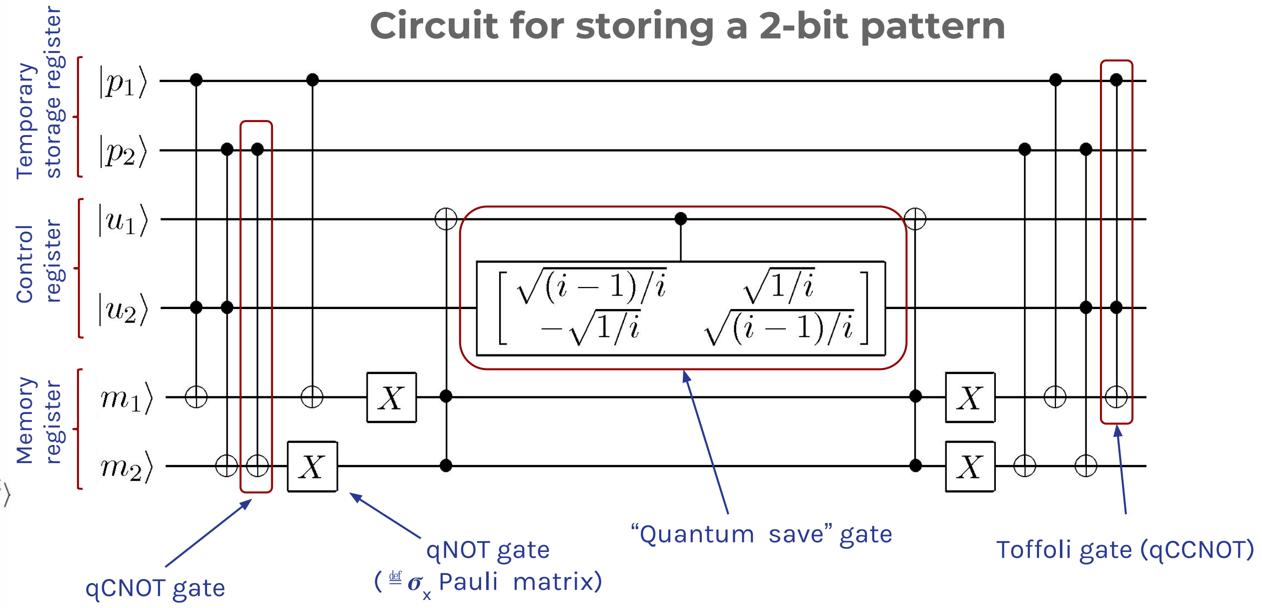
6. $|\psi_5^i\rangle = {}^{dc}\hat{X}_{m_1 \ldots m_d u_1} |\psi_4^i\rangle$

7. $|\psi_6^i\rangle = \prod_{j=d}^{1} {}^{1c}\hat{X}_{p_j^i m_j} \hat{X}_{m_j} |\psi_1^i\rangle$

$$= \frac{1}{\sqrt{p}} \sum_{k=1}^{i} |p^i; 00; p^k\rangle + \sqrt{\frac{p-i}{p}} |p^i; 01; p^i\rangle$$

8. $|\psi_7^i\rangle = \prod_{j=d}^{1} {}^{2c}\hat{X}_{p_j^i u_2 m_j} |\psi_6^i\rangle$

Circuit for storing a 2-bit pattern



[1] C.A Trugenberger, Probabilistic Quantum Memories. Phys Rev. Lett. Vol 87, 6 (2001)

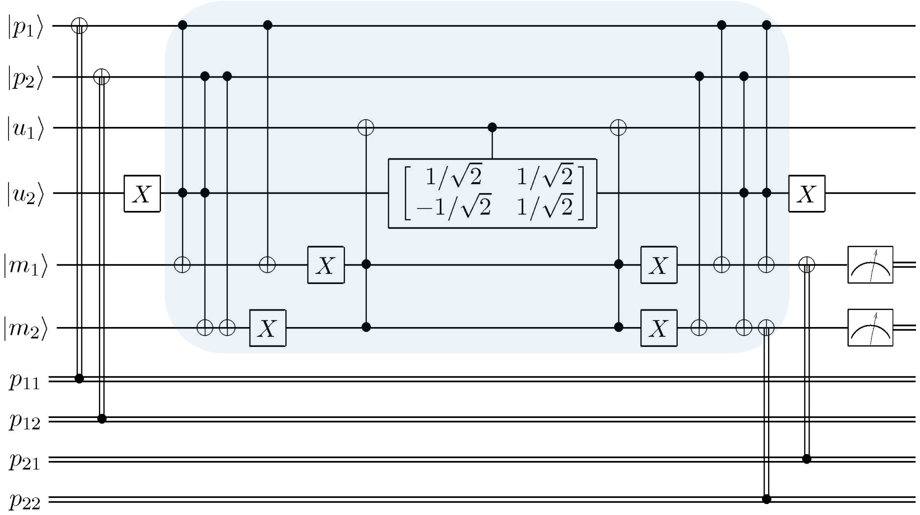
QuAM storage protocol

2-bit patterns example

The end-to-end circuit for storing two 2-bit patterns: “01” and “10”

“Empty” memory.

$$|00;00;00\rangle$$



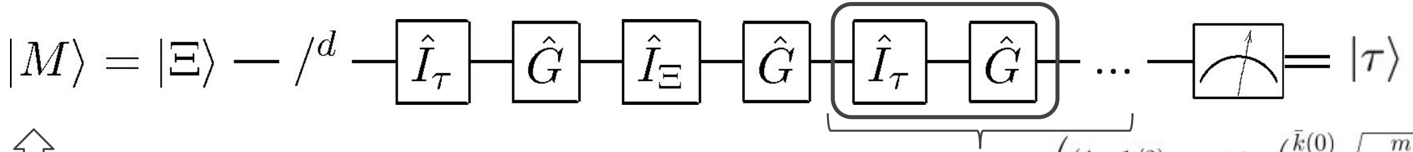
Classical bits for pattern read-in

Patterns ‘01’ and ‘10’ saved.

$$\frac{1}{\sqrt{2}}|01;00\rangle\left(|01\rangle+|10\rangle\right)$$

QuAM retrieval protocol

Generalized Grover's algorithm*



Grover's cycle. Repeated $T_j = NI\left(\frac{(j+1/2)\pi - \arctan\left(\frac{\bar{k}(0)}{\bar{l}(0)}\sqrt{\frac{m}{N-m}}\right)}{\arccos(1-\frac{2m}{N})}\right), \quad j = 0, 1, 2, \dots$ times.

$$\sum_{i=1}^{m} k_i(t)|x_i\rangle + \sum_{i=m+1}^{N} l_i(t)|x_i\rangle$$

States that match the target pattern.

States that don't match the target pattern.

* $\hat{I}_\tau$ - "quantum oracle" operator. Inverts the phase of state representing the target pattern τ.
$\hat{G}$ - Grover's diffusion operator. Inverts all amplitudes about the amplitudes average.
$\hat{I}_\Xi$ - Inverts phases of all terms originally present in memory.

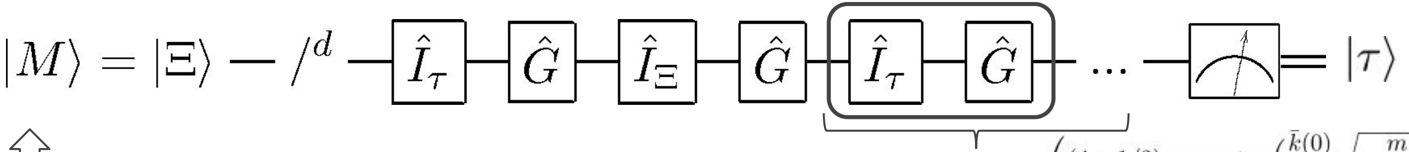
3.

Algorithmic properties

QuAM retrieval protocol

Generalized Grover's algorithm*

Analysis



Grover's cycle. Repeated $T_j = NI\left(\frac{(j+1/2)\pi - \arctan\left(\frac{\bar{k}(0)}{\bar{l}(0)}\sqrt{\frac{m}{N-m}}\right)}{\arccos(1-\frac{2m}{N})}\right)$, $j = 0, 1, 2, \ldots$ times.

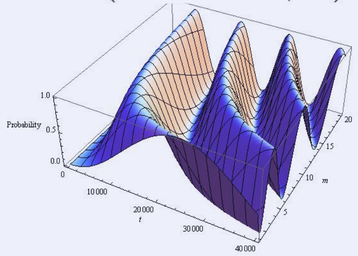
$$\sum_{i=1}^{m} k_i(t)|x_i\rangle + \sum_{i=m+1}^{N} l_i(t)|x_i\rangle$$

States that match the target pattern.

States that don't match the target pattern.

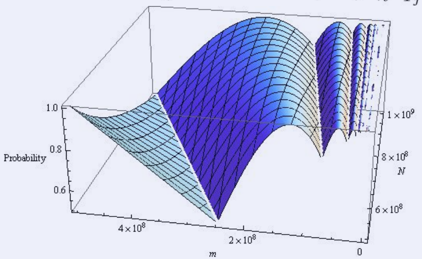
Probability "ramp-up" vs. pattern matches

$$P(t,m) = \sin^2\left((2t+1)\arcsin\sqrt{\frac{m}{N}}\right)\Bigg|_{N=10^9}$$



Peak probability vs. pattern matches and memory capacity

$$P(m,N) = \sin^2\left((2t+1)\arcsin\sqrt{\frac{m}{N}}\right)\Bigg|_{t=T_j}$$



$$m = 1, N = 10^9 : T_0 = 24836, P_{max} = 0.99999999999965568$$
$$m = 20, N = 10^9 : T_0 = 5553, P_{max} = 0.9999999991404647$$

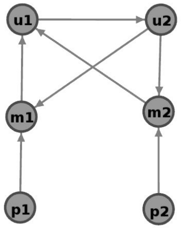
Note: neither quantum noise, nor probabilistic memory cloning operations, are taken into account here.

* $\hat{I}_\tau$ - "quantum oracle" operator. Inverts the phase of state representing the target pattern τ.
$\hat{G}$ - Grover's diffusion operator. Inverts all amplitudes about the amplitudes average.
$\hat{I}_\Xi$ - Inverts phases of all terms originally present in memory.

Topological complexity of QuAM[1]

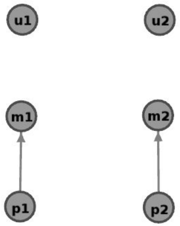
- **Storage** connectivity requirements

d=2



- **Retrieval** connectivity requirements

d=2



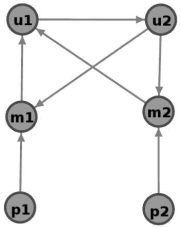
[1] **{p}**, **{u}** and **{m}** nodes represent qubits from temporary storage, control and memory registers.
d - pattern length

Topological complexity of QuAM[1]

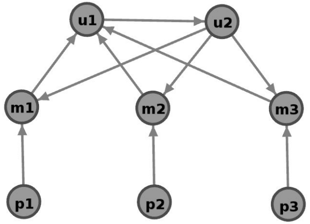
Analysis

- **Storage** connectivity requirements

d=2



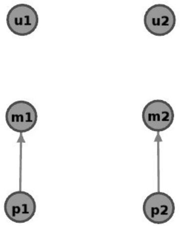
d=3



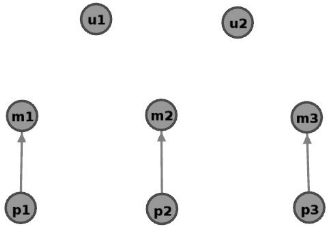
. . .

- **Retrieval** connectivity requirements

d=2



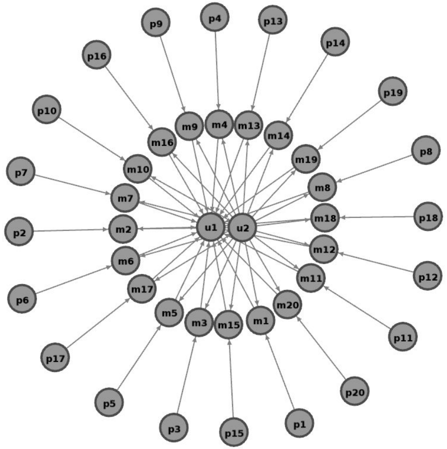
d=3



. . .

Cumulative QuAM requirements

d=20 (~ current pattern length in ATLAS)



Connectivity	*1*	*3*	*d+1*
Qubits	*d*	*d*	2

[1] **{p}**, **{u}** and **{m}** nodes represent qubits from temporary storage, control and memory registers.
d - pattern length

4.

Implementation

QuAM on QISKit

Quantum code

QISKit - Quantum Information Software Kit

An open source project comprising Python SDK, API and OpenQASM for implementing quantum algorithms on **IBM Quantum Experience (QE) hardware and simulators**.

Supported backends:

- **IBM QE cloud-based quantum chips**
  [5Q Sparrow/Raven, 16Q Albatross, 20Q]
- **Local/remote simulators**
  [with realistic noise models]

QuAM on QISKit

Quantum code

QISKit - Quantum Information Software Kit

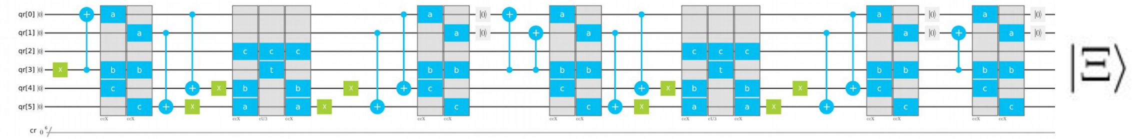
An open source project comprising Python SDK, API and OpenQASM for implementing quantum algorithms on **IBM Quantum Experience (QE) hardware and simulators**.

Supported backends:

- **IBM QE cloud-based quantum chips** [5Q Sparrow/Raven, 16Q Albatross, 20Q]
- **Local/remote simulators** [with realistic noise models]

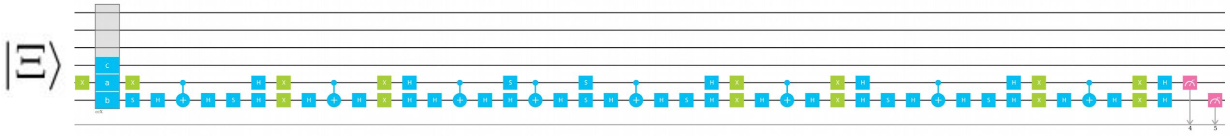
- **QuAM storage circuit generator [implemented]**

Ex.: complete circuit for encoding three 2-bit patterns



- **QuAM retrieval circuit generator [implemented]**

Ex.: complete circuit for retrieving one 2-bit pattern



Storage QASM

Snippet

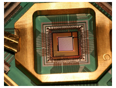
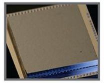
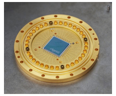
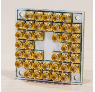
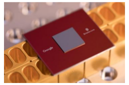
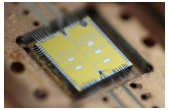
```
OPENQASM 2.0;
include "qelib1.inc";
qreg qr[6];
creg cr[6];
x qr[3];
cx qr[3],qr[0];
ccx qr[0],qr[3],qr[4];
ccx qr[1],qr[3],qr[5];
cx qr[1],qr[5];
cx qr[0],qr[4];
x qr[5];
x qr[4];
ccx qr[5],qr[4],qr[2];
cu3(1.23095941734077,3.14159265358979,3.14159265358979) qr[2],qr[3]
ccx qr[5],qr[4],qr[2];
x qr[5];
x qr[4];
cx qr[1],qr[5];
cx qr[0],qr[4];
ccx qr[0],qr[3],qr[4];
ccx qr[1],qr[3],qr[5];
reset qr[0];
reset qr[1];
cx qr[3],qr[0];
cx qr[3],qr[1];
```

Retrieval QASM

Snippet

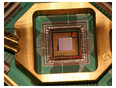
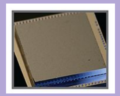
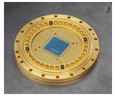
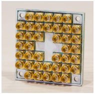
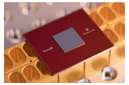
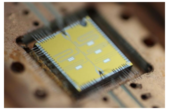
```
s qr[5];
h qr[5];
cx qr[4],qr[5];
h qr[5];
s qr[5];
h qr[4];
h qr[5];
x qr[4];
x qr[5];
h qr[5];
cx qr[4],qr[5];
h qr[5];
x qr[4];
x qr[5];
h qr[4];
h qr[5];
h qr[5];
cx qr[4],qr[5];
h qr[5];
```

Emerging Quantum Technologies

Quantum Chip	Qubits	Announced	Qubit Archetype	Computing Model
D-Wave 2000Q  	2048	01/2017	Superconducting **flux** qubits	Quantum annealing
IBM 20Q and 50Q  	20	11/2017	Superconducting **transmon** qubits	Quantum circuits
	50	11/2017 (tests)		
Rigetti 19Q 	19	12/2017	Superconducting **transmon** qubits	Quantum circuits
Intel Tangle Lake 	49	01/2018 (tests)	Superconducting qubits [1]	Quantum circuits
⟨G|oogl|e⟩ Bristlecone 	72	03/2018 (tests)	Superconducting **transmon** qubits	Quantum circuits
UC Berkeley QNL 	4 (8)	2017	Superconducting **transmon** qubits	Quantum circuits
	64	2022 ?		

[1] Archetype of superconducting qubits is not disclosed. Also investing in spin qubits in silicon.

Emerging Quantum Technologies

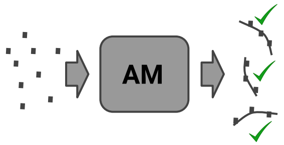
Quantum Chip	Qubits	Announced	Qubit Archetype	Computing Model
D-Wave 2000Q  	2048	01/2017	Superconducting **flux** qubits	Quantum annealing
IBM 20Q and 50Q  	20	11/2017	Superconducting **transmon** qubits	Quantum circuits
	50	11/2017 (tests)		
Rigetti 19Q 	19	12/2017	Superconducting **transmon** qubits	Quantum circuits
Intel Tangle Lake 	49	01/2018 (tests)	Superconducting qubits [1]	Quantum circuits
⟨G|oogl|e⟩ Bristlecone 	72	03/2018 (tests)	Superconducting **transmon** qubits	Quantum circuits
UC Berkeley QNL 	4 (8)	2017	Superconducting **transmon** qubits	Quantum circuits
	64	2022 ?		

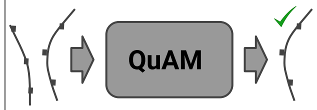
[1] Archetype of superconducting qubits is not disclosed. Also investing in spin qubits in silicon.

Challenges and Opportunities

Functional trade-offs

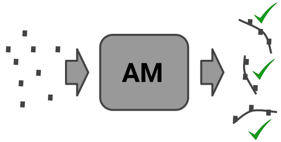
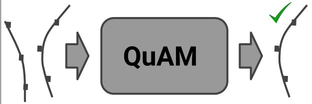
AM on ASICs **assembles** track patterns from hits:	QuAM **restores, validates** and/or **generalizes** track patterns:





Challenges and Opportunities

Functional trade-offs

AM on ASICs **assembles** track patterns from hits: 	QuAM **restores, validates** and/or **generalizes** track patterns: 

Memory persistence

Memory state collapses with each query. Repetitive re-initialization can be a show stopper. A possible solution may employ **probabilistic cloning of memory**.

Memory in real-time pattern matching

	RAM	Associative Memory (CAM)		
		ASICs	Hopfield networks	QuAM
Capacity[1] (patterns)	$\mathcal{O}(N/d)$	$\mathcal{O}(N/d)$	$\mathcal{O}(N)_{N\geq d}$: $\leq kN$, $0.138\leq k\leq 2$ [2]	$\mathcal{O}(2^d)$
Recall speed	$\mathcal{O}(N/d)$	$\mathcal{O}(1)$	$\Omega(N)_{N\geq d}$	$\mathcal{O}(\mathcal{O}(N)+N^{1/2})$
Spurious memories	None	None	Possible (e.g., $\mathcal{O}(e^N)$ for a Hebb-type rule[3])	None
Incomplete/noisy inputs	Yes/No	Yes/No	Yes/Yes	Yes/Yes
Power dissipation	Low	High	Low	Low
Nature	Det.	Det.	Deterministic	Probabilistic
Cost	Low	High	R&D	R&D

[1] **N** is the total number of elementary storage units in a device (memory cells/neurons), **d** - length of a pattern.
[2] B. Muller, J. Reinhardt, M.T. Strickland, Neural networks: An Introduction. Springer, 1995
[3] J. Bruck, V. P. Roychowdhury. On The Number of Spurious Memories in the Hopfield Model. IEEE Trans. on Information Theory, V. 36, 2, 1990

Summary

- **QC paradigm** can yield **asymmetrical advantages** in handling certain challenges of HE HEP real-time track pattern recognition
- **QuAM** features:
  - **Exponential storage capacity (absolute)**
  - ***Optimal*** QA for pattern **recall**
- Current status:
  - **Theoretical analysis** of QuAM properties **completed**
    - Memory initialization iterations
    - Recall probability bounds
    - Topological complexity analysis
  - **Storage/retrieval** quantum circuit generators **implemented** in QISKit
    - Ready to run on real quantum hardware
- Next steps:
  - Mitigate the memory initialization bottleneck
  - Scale up QuAM simulations to high-order patterns
  - Do full-stack performance tests (timing, efficiency)